Adapting Models to Signal Degradation using Distillation

Jong-Chyi Su
jcsu@cs.umass.edu
Subhransu Maji
smaji@cs.umass.edu

University of Massachusetts, Amherst
Amherst, MA 01003

Abstract

Model compression and knowledge distillation have been successfully applied for cross-architecture and cross-domain transfer learning. However, a key requirement is that training examples are in correspondence across the domains. We show that in many scenarios of practical importance such aligned data can be synthetically generated using computer graphics pipelines allowing domain adaptation through distillation. We apply this technique to learn models for recognizing low-resolution images using labeled high-resolution images, non-localized objects using labeled localized objects, line-drawings using labeled color images, etc. Experiments on various fine-grained recognition datasets demonstrate that the technique improves recognition performance on the low-quality data and beats strong baselines for domain adaptation. Finally, we present insights into workings of the technique through visualizations and relating it to existing literature.

1 Introduction

One of the challenges in computer vision is to build models for recognition that are robust to various forms of degradation of the quality of the signal such as loss in resolution, lower signal-to-noise ratio, poor alignment of the objects in images, etc. For example, the performance of existing models for fine-grained recognition drop rapidly when the resolution of the input image is reduced (see Table 1).

In many cases abundant high-quality data is available at training time, but not at test time. For example, one might have high-resolution images of birds taken by a professional photographer, while an average user might upload blurry images taken from their mobile devices for recognition. We propose a simple and effective way of adapting models in such scenarios. The idea is to *synthetically* generate data of the second domain from the first and *forcing agreement* between the model predictions across domains (Figure 1). The approach is a simple generalization of a technique called model compression, or knowledge distillation [2, 8, 22].

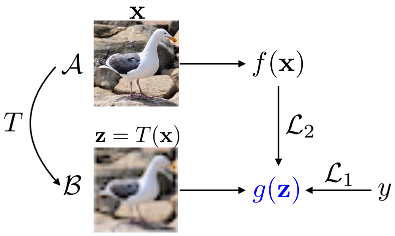


Figure 1: The objective of the CQD encourages agreement between $g(z)$ and $f(x)$ for each $z = T(x)$.

The main contribution of our work is to identify several practical scenarios where this idea can be applied. The simplest case is when domain $\mathcal{B}$ is a "degraded" version of domain $\mathcal{A}$. For example, when domain $\mathcal{B}$ has lower resolution than $\mathcal{A}$, or has no color information. It is easy to generate aligned data by applying known transformations T to obtain paired data of the form $[\mathbf{x}, T(\mathbf{x})]$. We also identify some non-trivial cases, such as when domain $\mathcal{A}$ has images of objects with known bounding-boxes while domain $\mathcal{B}$ does not. In such situations, a common approach is to train an object detector to localize the object and then classify the image. Our approach offers an alternate strategy where we first train a model on the cropped images and distill it to a model on full images. Experiments show that the improvements are significant, and in some cases matching the results using an object detector. Similarly, we can apply our technique to recognize distorted images as an alternative to Spatial Transformer Networks [25]. We call our approach Cross Quality Distillation (CQD).

We perform experiments on recognizing fine-grained categories of birds and cars using off-the-shelf Convolutional Neural Networks (CNNs). Experiments are performed on improving the recognition of low-quality data using high-quality data with various kinds of degradation (Figure 3). This is a challenging task even on the high-quality images, but performance of the models are often dramatically lower when directly applied on the low-quality images. Our experiments show that CQD leads to significant improvements over a model trained directly on the low-quality data and other strong baselines for domain adaptation, such as fine-tuning and "staged training" [38]. The model works across a variety of tasks and domains without any task-specific customization. Finally, we present insights into why the method works by relating it to the area of curriculum learning [3] and through visualizations of the learned models.

2 Related Work

Knowledge distillation The proposed approach is inspired by "knowledge distillation" technique [22] where a simple classifier g, *e.g.* a shallow neural network, is trained to imitate the outputs of a complex classifier f, *e.g.* a deep neural network (Figure 2a). Their experiments show that the simple classifier generalizes better when provided with the outputs of the complex classifier during training. This is based on an idea pioneered by Bucilă *et al.* [8] in a technique called "model compression" where simple classifiers such as linear models were trained to match the predictions of a decision-tree ensemble, leading to compact models. Thus, CQD can be seen as a generalization of model compression when the domains of the two classifiers $\mathcal{A}$ and $\mathcal{B}$ are different (Figure 2d). "Learning without forgetting" [30] shows that applying distillation on transfer learning can outperform fine-tuning, and has similar performance with multitask learning (joint training) but without using the data of original task. In this paper, we focus on domain adaptation problem where the tasks are the same but with paired data from different domains.

Learning with privileged Information The framework of learning with privileged information (LUPI) [48] (Figure 2b) deals with the case when additional information is available at training time but not at test time. The general idea is to use the side information to guide the training of the models. For example, the SVM+ approach [48] modifies the margin for each training example using the side information to facilitate the training on the input features. Most of these approaches require an explicit representation of the side information, i.e., the domain $\mathcal{A}$ can be written as a combination of domain $\mathcal{B}$ and side information domain

$$\begin{tikzcd}
\mathbf{x} \arrow[r] \arrow[d, dash, dotted] & f(\mathbf{x}) \arrow[d] & y \arrow[l] \arrow[d, dash, dotted] \\
\mathbf{x} \arrow[r] & g(\mathbf{x}) & y \arrow[l]
\end{tikzcd}$$

(a) model compression

$$\begin{tikzcd}
(\mathbf{z},\mathbf{s}) \arrow[d, dash, dotted] \arrow[dr] & & \\
\mathbf{z} \arrow[r] & g(\mathbf{z}) & y \arrow[l]
\end{tikzcd}$$

(b) LUPI

$$\begin{tikzcd}
\mathbf{x} \arrow[r] & f(\mathbf{x}) \arrow[d, "{y, y' \in \mathcal{Y}}"] & y \arrow[l] \\
\mathbf{z} \arrow[r] & g(\mathbf{z}) & y' \arrow[l]
\end{tikzcd}$$

(c) domain adaptation

$$\begin{tikzcd}
\mathbf{x} \arrow[r] \arrow[d, dash, dotted] & f(\mathbf{x}) \arrow[d] & y \arrow[l] \arrow[d, dash, dotted] \\
\mathbf{z} \arrow[r] & g(\mathbf{z}) & y \arrow[l]
\end{tikzcd}$$

(d) CQD

Figure 2: Illustration of the relationships between CQD and other techniques. An arrow points to the direction of variable dependency, and dotted lines denote that the variables are observed together. *(a) Model compression:* g is trained to mimic the outputs of f on the same input $\mathbf{x}$. *(b) LUPI:* g is trained with side information $\mathbf{s}$ observed together with input $\mathbf{z}$. *(c) Domain adaptation:* $\mathbf{x}$ and $\mathbf{z}$ are drawn independently from different domains but the tasks are the same, i.e. $y, y' \in \mathcal{Y}$. *(d) CQD:* can be seen as (i) a generalization of model compression where the inputs of the two functions are different, (ii) a specialization of domain adaptation when $\mathbf{z}$ can be synthesized from $\mathbf{x}$.

$\mathcal{S}$. For example, such models have been used to learn classifiers on images when additional information about them such as tags and attributes are available at training time. We note that Lopez-Paz *et al.* [35] made a similar observation unifying distillation and learning with privileged information.

Domain adaptation Techniques for domain adaptation addresses the performance loss due to domain-shift from training to testing, leading to degradation in performance. For example, visual classifiers trained on clutter-free images do not generalize well when applied to real-world images. Typically it is assumed that a large number of labeled examples exist for the source domain, but limited to no labeled data is available for the target domain. To increase feature generalization, some approaches [34, 45] minimize the domain discrepancy through Maximum Mean Discrepancy (MMD) [20]. Other approaches learn a parametric transformation to align the representations of the two domains [17, 28, 40, 44]. Bousmalis *et al.* [5] combines encoder-decoder structure and different loss functions to learn shared and domain-specific features explicitly. Ganin *et al.* [18] proposed the domain-adversarial neural networks (DANN) which learns representations by competing with an adversarial network trained to discriminate the domains. Instead of learning domain-invariant features, some approaches [6, 32, 47] use Generative Adversarial Networks (GANs) to generate images of target domain for unsupervised domain adaptation.

When some labeled data is available for the target domain (supervised case), methods for multi-task learning [9] are also applicable, including ones that are "frustratingly easy" [13]. CQD is a special case of supervised domain adaptation where we have correspondence between samples from the source and target domain, i.e., in supervised domain adaptation we have training data of the form $(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathcal{A}$ and $(\mathbf{z}_j, y_j), \mathbf{z}_j \in \mathcal{B}$, where $\mathbf{x}_i$ and $\mathbf{z}_j$ are drawn independently from the source and target domain respectively, and $y_i, y_j \in \mathcal{Y}$. In CQD we know that $\mathbf{x}_i$ and $\mathbf{z}_i$ are two views of the same instance. This provides richer information to adapt models across domains. Our experiments show that distillation leads to greater improvements in accuracy compared to fine-tuning, a commonly used approach for domain adaptation, and "staged training" [38], specifically designed for scenarios like ours where high-quality data is available at training time. The idea of transferring task knowledge through distillation has been applied for simultaneous domain adaptation and task transfer by Tzeng *et al.* [46]. They tried to match the average predicted label scores across examples in source domain to that of the target domain as instances lack one-to-one correspondence. In contrast, paired data in CQD allows matching of label distributions on per-instance basis.

Cross modal learning When multiple modalities of images are simultaneously available, the information about one domain can guide representation learning for another domain. Recent works have used a similar idea to ours to learn representations of depth from color using RGB-D data [21, 23], representations of video from ambient sound [37] and vice-versa [1], as well as visual representations through self-supervised colorization [29, 54]. Our work identifies several novel situations when distillation can be applied effectively. For example, we train a model to recognize distorted images of birds by distilling a model trained on non-distorted ones.

3 CQD Framework

Assume that we have data in the form of $(\mathbf{x}_i, \mathbf{z}_i, y_i)$, $i = 1, 2, \ldots, n$ where $\mathbf{x}_i \in \mathcal{A}$ is the high-quality data, $\mathbf{z}_i \in \mathcal{B}$ is the corresponding low-quality data, and $y_i \in \mathcal{Y}$ is the target label. In practice only the high-quality data $\mathbf{x}_i$ is needed since $\mathbf{z}_i$ can be generated from $\mathbf{x}_i$. The idea of CQD is to first train a model f to predict the labels on the high-quality data and train a second model g on the low-quality data by forcing an agreement between their corresponding predictions by minimizing the following objective (Fig. 1):

$$\sum_{i=1}^{n} \mathcal{L}_1 \left(g(\mathbf{z}_i), y_i\right) + \lambda \sum_{i=1}^{n} \mathcal{L}_2 \left(g(\mathbf{z}_i), f(\mathbf{x}_i)\right) + \mathcal{R}(g). \tag{1}$$

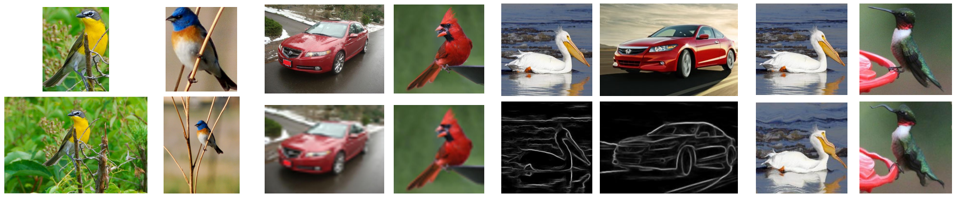
Here, $\mathcal{L}_1$ and $\mathcal{L}_2$ are loss functions, λ is a trade-off parameter, and $\mathcal{R}(g)$ is a regularization term. The intuition for this objective is that by imitating the prediction of f on the high-quality data g can learn to generalize better on the low-quality data.

All our experiments are on multi-class classification datasets and we model both f and g using multi-layer CNNs, pre-trained on ImageNet dataset, with a final softmax layer to produce class probabilities $\mathbf{p} = \sigma(\mathbf{z})$, i.e., $p_k = e^{z_k} / \sum_j e^{z_j}$. We use the cross-entropy loss $\mathcal{L}_1(\mathbf{p}, \mathbf{q}) = \sum_i q_i \log p_i$, and the cross-entropy of the predictions smoothed by a temperature parameter T for $\mathcal{L}_2(\mathbf{p}, \mathbf{q}) = \mathcal{L}_1 \left(\sigma\left(\log(\mathbf{p})/T\right), \sigma\left(\log(\mathbf{q})/T\right)\right)$. When $T = 1$, this reduces to the standard cross-entropy loss. We also found that squared-error between the logits ($\mathbf{z}$) worked similarly. More details can be found in the experiments section.

4 Experiments

We begin by describing datasets, models, and training protocols used in our experiments. Section 4.1 describes the results of various experiments on CQD. Section 4.2 describes experiments for simultaneous quality distillation and model compression. Finally, Section 5 visualizes the distilled models to provide an intuition of why and how distillation works.

Datasets We perform experiments on the CUB 200-2011 dataset [50] consisting of 11,788 images of 200 different bird species, and on the Stanford cars dataset [26] consisting of 16,185 images of 196 cars of different models and makes. Classification requires the ability to recognize fine-grained details which is impacted when the quality of the images is poor. Using the provided images and bounding-box annotations in these datasets, we create several cross-quality datasets which are described in detail in the Section 4.1 and visualized in Figure 3. We use the training and test splits provided in the datasets.



(a) localized → non localized (b) high → low resolution (c) color → edges (d) regular → distorted

Figure 3: Examples images from various cross-quality datasets used in our experiments. Images are from the birds [50] and cars dataset [26]. In each panel, the top row shows examples of the high-quality images and the bottom row shows examples of the corresponding low-quality images. These include (a) localized and non-localized images, (b) high- and low-resolution images, (c) color and edge images, and (d) regular and distorted images.

Models In our experiments, both f and g are based on CNNs pre-trained on the ImageNet dataset [14]. In particular we use `vgg-m` [10] and `vgg-vd` models [42] which obtain competitive performance on the ImageNet dataset. While there are better performing models for these tasks, *e.g.* those using novel model architectures[12, 25, 31, 41], and using additional annotations to train part and object detectors [4, 7, 52, 53], we perform experiments with simple models in the interest of a detailed analysis. However, we believe that our method is general and can be applied to other recognition architectures as well.

Methods Below we describe various methods used in our experiments:

1. **Train on $\mathcal{A}$:** Starting from the ImageNet pre-trained model, we replace the 1000-way classifier (last layer) with a k-way classifier initialized randomly and then fine-tune the entire model with a small learning rate on domain $\mathcal{A}$. This is a standard way of transfer learning using deep models, and has been successfully applied for a number of vision tasks including object detection, scene classification, semantic segmentation, texture recognition, and fine-grained classification [11, 16, 19, 31, 33, 36, 39].

2. **Train on $\mathcal{B}$:** Here we fine-tune the ImageNet pre-trained model on domain $\mathcal{B}$.

3. **Train on $\mathcal{A}+\mathcal{B}$:** Here we fine-tune the model on domain $\mathcal{A}$ combined with domain $\mathcal{B}$. Data augmentation is commonly used while training CNNs to make them more robust.

4. **Train on $\mathcal{A}$, then train on $\mathcal{B}$:** This is a combination of $\mathcal{A}$ and $\mathcal{B}$ where the fine-tuning on domain $\mathcal{B}$ is initialized from the model fine-tuned on domain $\mathcal{A}$. This "staged training" was recently proposed in [38] as a state-of-the-art technique for low-resolution image recognition. However, this method can only be applied when both f and g have the same structure. This is denoted by $\mathcal{A},\mathcal{B}$ in our experiments.

5. **Cross quality distillation (CQD):** Here we use a model f trained on domain $\mathcal{A}$ (Method 1) to guide the learning of a second model g on domain $\mathcal{B}$ using CQD (Equation 1). Like before, when f and g have identical structure we can initialize g from f instead of the ImageNet model with random weights for the last layer.

Optimization details There are two parameters, T and λ, in the CQD model. The optimal value we found on validation set is $T = 10$ for all experiments, and $\lambda = 200$ for the CUB, $\lambda = 50$ for the CARS dataset. The optimization in Equation 1 was solved using batch

Description	Method	Test	Local.	Resolution		Edge		Dist.	Local. + Res.
			CUB	CUB	CARS	CUB	CARS	CUB	CUB
Upper bound	$\mathcal{A}$	$\mathcal{A}$	67.0	67.0	59.3	67.0	59.3	67.0	67.0
No adaptation	$\mathcal{A}$	$\mathcal{B}$	57.4	39.4	7.6	1.9	4.2	49.7	24.9
Fine-tuning	$\mathcal{B}$	$\mathcal{B}$	60.8	61.0	41.6	29.2	45.5	58.4	46.2
Data augment.	$\mathcal{A}+\mathcal{B}$	$\mathcal{B}$	63.6	62.2	47.3	32.5	51.3	61.7	51.7
Staged training	$\mathcal{A}$,$\mathcal{B}$	$\mathcal{B}$	62.4	62.3	48.4	30.4	50.1	60.9	50.4
Proposed	CQD	$\mathcal{B}$	**64.4**	**64.4**	**48.8**	**34.1**	**51.5**	**63.0**	**52.7**

Table 1: **Cross quality distillation results.** Per-image accuracy on birds dataset (CUB) [50] and Stanford cars dataset (CARS) [26] for various methods and quality losses. All results are using $f = g =$ vgg-m model. Training on $\mathcal{A}$ and testing on $\mathcal{A}$ is the upper bound of the performance in each setting (top row). Training on $\mathcal{A}$ and testing on $\mathcal{B}$ (no adaptation) often leads to a significant loss in performance. The proposed technique (CQD) outperforms fine-tuning ($\mathcal{B}$), data augmentation ($\mathcal{A}$ + $\mathcal{B}$), and staged training ($\mathcal{A}$,$\mathcal{B}$) [38] on all datasets.

stochastic gradient descent, with learning rate starting from `0.0005` (`0.0005` for CUB, `0.001` for CARS) changing linearly to `0.00005` after 30 epochs. Other parameters are as follows: `momentum=0.9, weight decay=0.0005, batch size=128` (=32 when training `vgg-vd`). Instead of cross-entropy we also tried squared-distance on the logits **z** as the loss function [2]. There was no significant difference between the two and we used cross-entropy for all our experiments. Our implementation is based on MatConvNet [49].

4.1 Cross Quality Distillation Results

We experiment with five different kinds of quality reduction to test the versatility of the approach. For each case we report per-image accuracy on the test set provided in the dataset. Results using the `vgg-m` model for both function f and g are summarized in Table 1 and are described in detail below. The main conclusions are summarized in the end of this section.

4.1.1 Localized to Non-localized Distillation

To create the high-quality data, we use the provided bounding-boxes in the CUB dataset to crop the object in each image. In this dataset, birds appear in various locations and scales and in clutter. Therefore, `vgg-m` trained and evaluated on the localized data obtains 67.0% accuracy, but when applied the non-localized data obtains only 57.4% accuracy (Table 1). When the model is trained on the non-localized data the performance improves to 60.8%. Staged training $\mathcal{A}$,$\mathcal{B}$ improves the performance to 62.4%, but CQD improves further to 64.4%.

For this task another baseline would be to train an object detector which first localizes the objects in images. For example, Krause *et al.* [27] report around 2.6% drop in accuracy (67.9% → 65.3%) when a R-CNN based object detector is used to estimate bounding-boxes of objects at test time instead of using true bounding-boxes (Table 2 in [27], CNN+GT BBox+ft vs. R-CNN+ft). Remarkably, using CQD we observe only 2.6% drop in performance (67.0% → 64.4%) without running any object detector. Moreover, our method only requires a single CNN evaluation and hence is faster. In Section 5 we provide insights into why the distilled model performs better on non-localized images.

4.1.2 High to Low Resolution Distillation

Here we evaluate how models perform on images of various resolutions. For the CUB dataset we use the localized images resized to 224 × 224 for the high-resolution images, downsample to 50 × 50, and upsample to 224 × 224 again for the low-resolution images. For the CARS dataset we do the same but for the entire image (bounding-boxes are not used).

The domain shift leads to large loss in performance here. On CUB the performance of the model trained on high-resolution data goes down from 67.0% to 39.4%, while the performance loss on CARS is even more dramatic going from 59.3% to a mere 7.6%. Man-made objects like cars contain high-frequency details such as brand logos, shapes of head-lights, etc., which are hard to distinguish in the low-resolution images. A model trained on the low-resolution images does much better, achieving 61.0% and 41.6% accuracy on birds and cars respectively. Color cues in the low-resolution are much more useful for distinguishing birds than cars which might explain the better performance on birds. Using CQD the performance improves further to 64.4% and 48.8% on the low-resolution data. On CARS the effect of both staged training and CQD is significant, leading to more than 7% boost in performance.

4.1.3 Color to Edges Distillation

Recognizing line-drawings can be used for retrieval of images and 3D shapes using sketches and has several applications in search and retrieval. As a proxy for line-drawings, we test the performance of various methods on edge images obtained by running the structured edge detector [15] on the color images. In contrast to low-resolution images, edge images contain no color information but preserve most of the high-frequency details. This is reflected in the better performance of the models on CARS than CUB dataset (Table 1). Due to the larger domain shift, a model trained on color images performs poorly on edge images, obtaining 1.9% and 4.2% accuracy on CUB and CARS receptively.

Using CQD the performance improves significantly from 45.5% to 51.5% on CARS. On the CUB dataset the performance also improves from 29.2% to 34.1%. The strong improvements on recognizing line drawings using distillation and staged training suggests that a better strategy to recognize line drawings of shapes used in various sketch-based retrieval applications [43, 51] is to first fine-tune the model on realistically rendered 3D models (*e.g.* with shading and texture) then distill the model to edge images.

4.1.4 Non-distorted to Distorted Distillation

Here the high-quality dataset is the localized bird images. To distort an image as seen in Figure 3d, we use the thin plate spline transformation with uniform grid of 14×14 control points. Each control point is mapped from a regular grid to a point randomly shifted by Gaussian distribution with zero mean and 4 pixels variance. Recognizing distorted images is challenging, and the performance of a model trained and evaluated on such images is 8.6% worse (67.0% → 58.4%). Using CQD the performance improves from 58.4% to 63.0%.

On this dataset a baseline would be to remove the distortion by alignment methods such as congealing [24], or use a model that estimates deformations during learning, such as spatial transformer networks [25]. These methods are likely to work well but they require the knowledge of the space of transformations and are non-trivial to implement. On the other hand, CQD is able to nearly halve the drop in performance of the same CNN model without any knowledge of the nature of distortion and is easy to implement. Thus, CQD may be used whenever we can model the distortions algorithmically. For example, computer graphics techniques can be used to model the distortions from underwater imaging.

4.1.5 Color to Non-localized and Low Resolution Distillation

Here the images has two different degradations at the same time: the low-quality data is low resolution images with the object in clutter, where the high-quality data is high resolution images cropped by the bounding boxes provided in the CUB dataset. Without adaptation, the performance drops to 24.9%, more than when only have one type of degradation (57.4% and 39.4% separately). We want to stress that the type of degradation in domain $\mathcal{B}$ can be arbitrary, as long as we have the instance-level correspondence between different domains which can be done by applying known transformations. As shown in the last column of Table 1, CQD improves 6.5% (46.2% → 52.7%) over fine-tuning.

Summary In summary we found that domain adaptation is critical since the performance of models trained on high-quality data is poor on the low-quality data. Data augmentation ($\mathcal{A}+\mathcal{B}$) and staged training ($\mathcal{A},\mathcal{B}$) are quite effective, but CQD provides better improvements suggesting that adapting models on a per-example basis improves knowledge transfer across domains. CQD is robust and only requires setting a handful of parameters, such as T and λ, across a wide variety of quality losses. In most cases, CQD cuts the performance gap between the high- and low-quality data in half.

4.2 Simultaneous CQD and Model Compression

In this section we experiment if a deeper CNN trained on high-quality data can be distilled to a shallow CNN trained on the low-quality data. This is the most general version of CQD where both the domains and functions f,g change. The formulation in Equation 1 does not require f and g to be identical. However, $\mathcal{A}+\mathcal{B}$ and $\mathcal{A},\mathcal{B}$ baselines cannot be applied here.

We perform experiments on the CUB dataset using localized and non-localized images described earlier. The deeper CNN is the sixteen-layer "very deep" model (`vgg-vd`) and the shallow CNN is the five-layer `vgg-m` model used in the experiments so far. The optimal parameters obtained on the validation set for this setting were $T = 10, \lambda = 50$.

The results are shown in Table 2. The first row contains results using CQD for `vgg-m` model which are copied from Table 1 for ease of comparison. The third row shows the same results using the `vgg-vd` model. The accuracy is higher across all tasks. CQD leads to an improvement of 2.9% (69.5% → 72.4%) for the deeper model. The middle row shows results for training the `vgg-m` model on non-localized images from a `vgg-vd` model trained on the localized images. This leads to a further improvement of 0.9% (63.7% → 64.6%) suggesting that model compression and cross quality distillation can be seamlessly combined.

	training → testing		
$f \rightarrow g$	$\mathcal{A} \rightarrow \mathcal{A}$	$\mathcal{B} \rightarrow \mathcal{B}$	CQD $\rightarrow \mathcal{B}$
`vgg-m` → `vgg-m`	67.0	60.8	63.7
`vgg-vd` → `vgg-m`	-	-	64.6
`vgg-vd` → `vgg-vd`	74.9	69.5	72.4

Table 2: Accuracy of various techniques on the CUB localized/non-localized dataset.

5 Understanding Why CQD Works

Relation to curriculum learning Curriculum learning is the idea that models generalize better when training examples are presented in the order of their difficulty. Bengio *et al.* [3] showed a variety of experiments where non-convex learners reach better optima when more difficult examples are introduced gradually. In one experiment a neural network was trained to recognize shapes. There were two kinds of shapes: BasicShapes which are canonical circles, squares, and triangles, and GeomShapes which are affine distortions of the BasicShapes on more complex backgrounds. When evaluated only on test set of GeomShapes, the model first trained on BasicShapes then fine-tuned on GeomShapes, performed better than the model only trained on GeomShapes, or the one trained with a random ordering of both types of shapes.

We observe a similar phenomenon when training CNNs on low-quality data. For example, on the CARS dataset, staged training [38] $\mathcal{A}$,$\mathcal{B}$ outperforms the model trained on low-resolution data $\mathcal{B}$, when evaluated on the low-resolution data $\mathcal{B}$ (48.4% vs. 41.6%). Since low-quality data is more difficult to recognize, introducing them gradually might explain the better performance of the staged training and CQD techniques. Additional benefits of CQD come from the fact that paired high- and low-quality images allowing better knowledge transfer through distillation.

Understanding CQD through gradient visualizations Here we investigate how the knowledge transfer occurs between a model trained on localized images and non-localized images. Our intuition is that by trying to mimic the model trained on the localized images a model must learn to ignore the background clutter. In order to verify this, we compute the gradient of log-likelihood of the true label of an image with respect to the image using the CQD model and $\mathcal{B}$ model, both are trained only on non-localized images. Figure 4-left shows the gradients for two different images. The darkness of each pixel i is proportional to the norm of the gradient vector at that pixel $||G_i||_2$, $G_i = [G_i^r, G_i^g, G_i^b]$ for r, g, b color channels. The gradients of the CQD model are more contained within the bounding-box of the object, suggesting a better invariance to background clutter. As a further investigation we compute the fraction of gradients within the box: $\tau = (\sum_{i \in \text{box}} ||G_i||_2) \ / \ (\sum_{i \in \text{image}} ||G_i||_2)$. This ratio is a measure of how localized the relevant features are within the bounding-box. A model based on a perfect object detector will have $\tau = 1$. We compute τ for 1000 images for both CQD and $\mathcal{B}$ models and visualize them on a scatter plot as seen in Figure 4-right. On average the CQD model has higher τ than $\mathcal{B}$ model, confirming our intuition that the CQD model is implicitly able to localize objects.

6 Conclusion

We proposed a simple generalization of distillation, originally used for model compression, for cross quality model adaptation. We showed that CQD achieves superior performance than domain adaption techniques such as fine-tuning on a range of tasks, including recognizing low-resolution images, non-localized images, edge images, and distorted images. Our experiments suggest that recognizing low-quality data is a challenge, but by developing better techniques for domain adaptation one can significantly reduce the performance gap between the high- and low-quality data. We presented insights into why CQD works by relating it to various areas in machine learning and by visualizing the learned models.

Training highly expressive models with limited training data is challenging. A common strategy is to provide additional annotations to create intermediate tasks that can be easily solved. For example, annotations can be used to train part detectors to obtain pose, viewpoint, and location-invariant representations, making the fine-grained recognition problem easier. However, these annotation-specific solutions do not scale as new types of annotations

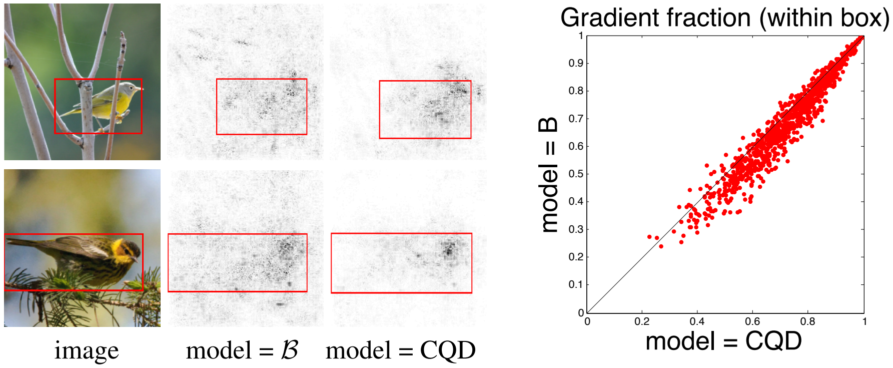


Figure 4: **Left:** Image and gradient of the image with respect to the true class label for the model trained on $\mathcal{B}$ (non-localized images) and CQD (from a model trained on localized images). Darker pixels represent higher gradient value. The gradients of the model trained using CQD are more focused on the foreground object. **Right:** The scatter plot of the fraction of total gradient within the bounding-box for 1000 training images for the two models.

become available. An alternate strategy is to use CQD by simply treating these annotations as additional features, learning a classifier in the combined space of images and annotations, and then distilling it to a model trained on images only. This strategy is much more scalable and can be easily applied as new forms of side information, such as additional modalities and annotations, become available over time. In future work, we aim to develop strategies for distilling deep models trained from richly-annotated training data for better generalization from small training sets.

Acknowledgement: This research was supported in part by the NSF grants IIS-1617917 and ABI-1661259, and a faculty gift from Facebook. The experiments were performed using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Tech Collaborative and GPUs donated by NVIDIA.

References

[1] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, 2009.

[4] L. Bourdev, S. Maji, and J. Malik. Describing People: Poselet-Based Approach to Attribute Classification. In *International Conference on Computer Vision (ICCV)*, 2011.

[5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[6] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. In *British Machine Vision Conference (BMVC)*, 2014.

[8] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[9] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[10] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference (BMVC)*, 2014.

[11] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] Hal Daumé III. Frustratingly easy domain adaptation. *ACL*, 2007.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] Piotr Dollár and C. Lawrence Zitnick. Structured forests for fast edge detection. In *International Conference on Computer Vision (ICCV)*, 2013.

[16] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, Ning Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2013.

[17] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, 2013.

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17(59), 2016.

[19] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[20] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13(Mar), 2012.

[21] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] G. Hinton, O. Vinyals, and J Dean. Distilling knowledge in a neural network. In *Deep Learning and Representation Learning Workshop, NIPS*, 2014.

[23] Judy Hoffman, Saurabh Gupta, Jian Leong, Sergio Guadarrama, and Trevor Darrell. Cross-modal adaptation for rgb-d detection. In *International Conference in Robotics and Automation (ICRA)*, 2016.

[24] Gary Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[25] Max Jaderberg, Karen Simonyan, and Andrew Zisserman. Spatial transformer networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013.

[27] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *International Conference on Computer Vision (ICCV)*, 2015.

[28] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[29] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[30] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[31] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. *International Conference on Computer Vision (ICCV)*, 2015.

[32] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] Mingsheng Long, Yue Cao, and Jianmin Wang. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015.

[35] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

[36] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[37] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *European Conference on Computer Vision (ECCV)*, 2016.

[38] Xingchao Peng, Judy Hoffman, Stella X. Yu, and Kate Saenko. Fine-to-coarse knowledge transfer for low-res image classification. In *IEEE International Conference on Image Processing (ICIP)*, 2016.

[39] A. Sharif Razavin, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *DeepVision workshop*, 2014.

[40] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*. Springer, 2010.

[41] Pierre Sermanet, Andrea Frome, and Esteban Real. Attention for fine-grained categorization. *International Conference on Learning Representation Workshop*, 2015.

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[43] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *International Conference on Computer Vision (ICCV)*, 2015.

[44] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[45] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[46] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *International Conference on Computer Vision (ICCV)*, 2015.

[47] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[48] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.

[49] Andrea Vedaldi and Karel Lenc. Matconvnet – convolutional neural networks for matlab. *Proceeding of the ACM International Conference on Multimedia*, 2015.

[50] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, CalTech, 2011.

[51] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[52] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *European Conference on Computer Vision (ECCV)*, 2014.

[53] Ning Zhang, Manohar Paluri, Marc'Aurelio Rantazo, Trevor Darrell, and Lubomir Bourdev. PANDA: Pose Aligned Networks for Deep Attribute Modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[54] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016.